NdProperties: Encoding Contexts in RDF Predicates with Inference Preservation

José M. Giménez-García[1] and Antoine Zimmermann[2]

[1] Univ Lyon, UJM-Saint-Étienne, CNRS, Laboratoire Hubert Curien UMR 5516, F-42023 Saint-Étienne, France
jose.gimenez.garcia@univ-st-etienne.fr

[2] Mines Saint-Étienne, Univ Lyon, Univ Jean Monnet, IOGS, CNRS, UMR 5516 LHC, Institut Henri Fayol, F - 42023 Saint-Étienne France
antoine.zimmermann@emse.fr

Abstract. Annotating RDF triples within a context (such as provenance) is becoming more and more important in the recent years. A number of approaches exist to represent this annotations, but how to distinguish the information of each context and be able to reason with it is still a challenge. In this work, we present formalizations of the necessary transformations in the RDF graph to encode the contextual information as contextualization function in RDF, and describe the existing approaches according to this formalization. We then present NdProperties, an OWL-DL ontology inspired by NdFluents and singleton properties, and present alternative contextualization functions for it.

Keywords: RDF · OWL · Contexts · Reasoning · Reification

1 Introduction

In the recent years, the necessity of annotating RDF statements with contextual information have been increasing. However, a triple can only represent a binary relations between a subject and an object. To make explicit the relationship between the statement and its context, either the information must be encoded outside of the RDF syntax, or the statement and its context must be split into multiple triples, possibly resulting in the subject and the object being disconnected.

With most common approaches for representing contextual information, inferences that were following from the original statement are not preserved. Recent approaches allow for some inferences to be preserved. However, the result of the inference has no connection with the context. This makes impossible to separate the knowledge in different context, and can lead to data that is inconsistent or that does not correspond with the ground truth, due to some undesirable inference results.

In this work we present *NdProperties*, an OWL-DL ontology to use triple predicates to relate statements to their context. This approach is inspired by Singleton Properties [8] (which also use the predicates to represent the statement) and NdFluents [5] (which is able to better separate the contexts), NdProperties can be seen as a concrete implementation of a generic approach where a subset of the terms is contextualized, explored also for all the terms in NdTerms [10]. The NdProperties ontology can be implemented in different ways, in order to allow more or less inferences.

The rest of the document is organized as follows: Section 2 presents some definitions, necessary to understand the rest of the work; Section 3 present current approaches to annotate statements with additional data; Section 4 describes the NdProperties ontology, with Section 5 studying its inference capabilities; finally, we provide some discussion possible future work in Section 6.

2 Preliminaries

In order to go from a set of triples, plus the desired contextual annotations, to a set of contextualized triples (*i.e.*, a set of reified or modified triples, plus additional triples that express the annotations), it is necessary to perform a transformation. In this section we introduce the notation we will use for the rest of the paper, formalize this transformation as a *contextualization function*, and present some desired properties that such a function should have. This section summarizes and updates previous work in the topic [5, 10].

We assume infinite disjoint sets $\mathcal{I}$ (IRIs), $\mathcal{B}$ (blank nodes), and $\mathcal{L}$ (literals). An RDF triple is a tuple $(s, p, o) \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$, where s is called the *subject*, p is the *predicate* and o is the *object*. We write $\mathcal{T}$ the infinite set of triples. An RDF graph G is a set of RDF triples. For the sake of simplicity, we will assume that a contextual annotation is a single IRI. We assume an infinite set of contexts $\mathcal{C} \subset \mathcal{I}$.

Definition 1 (Contextualization function). *A contextualization function in RDF is any function $f : \mathcal{C} \times 2^{\mathcal{T}} \to 2^{\mathcal{T}}$.*

Usually, contextualization functions need to create fresh terms that will be used to reify the statements. This is often done by renaming existing terms using one or more injective functions ren, which may or may not depend on the context or the type or term being renamed. This function is in practice usually arbitrary. The idea of a contextualization function is that it expresses that the knowledge of the original graph holds in the context. However, many meaningless contextualization functions are possible that do not necessarily encode that knowledge. In order to identify useful contextualization functions, we propose the following property:

Definition 2 (Inference preservation). *A contextualization function f* preserves inferences *wrt a set of graphs $\mathcal{G}$ iff for all $c \in \mathcal{C}$ and all graphs $G, G' \in \mathcal{G}$, iff $G \models G'$ then $f(c, G) \models f(c, G')$.*

Inference preservation expresses the idea that knowledge that can be inferred in the original graph should also be inferred within the context. It is also possible that the contextualization function preserves some inferences, but it is not able to encapsulate the inferred knowledge in the context. We call this property *Non-contextual Inference preservation*. Although this property can be useful if the goal of contextualization function is to simply annotate triples that are considered universally true, it can lead to undesirable inferences if that is not the case, as we will see in Section 5.

Definition 3 (Non-contextual Inference preservation). *A contextualization function f preserves entailments wrt a set of ontologies $\mathcal{O}$ iff for all $c \in \mathcal{C}$ and all ontologies $O, O' \in \mathcal{O}$, iff $O \models O'$ then $f(c, O) \models O$.*

This definitions will be used to study existing reification approaches, as well as the newly defined *NdProperties*.

3 Existing approaches for representing context

In this section we present and describe the most relevant reification approaches (namely RDF reification, n-ary relations, the singleton and companion properties, and NdFluents), and formalize their contextualization functions. In the following definitions, we will use t as the parameter of the ren function when the fresh term does not depend on a concrete term of the triple. Note that for RDF reification and n-ary relations more than one contextualization function could be possible. For the sake of simplicity, we consider an elementary transformation were the added term will be used to represent the context.

RDF reification [2, Sec. 5.3] is the standard W3C model to represent information about a statement. A contextualization function for reification can be defined as follows.

Definition 4 (RDF reification contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_R such that $f_R(c, G) = \{(\mathsf{ren}_c(t), type, Statement), (\mathsf{ren}_c(t), subject, s), (\mathsf{ren}_c(t), predicate, p), (\mathsf{ren}_c(t), object, o) \mid (s, p, o) = t \in G\}$.*

Techniques for representing n-ary relations in RDF [9] were published in 2006 as a W3C note. A contextualization function for them can be defined as follows.

Definition 5 (N-ary relations contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{nary} such that $f_{nary}(c, G) = \{(s, \mathsf{ren}^s_c(p), \mathsf{ren}^r_c(p)), (\mathsf{ren}^r_c(p), \mathsf{ren}^v_c(p), o) \mid (s, p, o) = t \in G\}$.*

Singleton properties [8] are a recent proposal based on creating a unique property for each triple and using it to reify the triple. It extends the RDF semantics in order to make each singleton property unique and to include its extension in the extension of the original property. However, these semantics can be emulated with the following contextualization function.

Definition 6 (Singleton Property contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{sp} such that $f_{sp}(c, G) = \{(s, \mathsf{ren}(p), o), (\mathsf{ren}(p), type, SingletonProperty), (\mathsf{ren}(p), singletonPropertyOf, p), (\mathsf{ren}(p), type, FunctionalProperty), (\mathsf{ren}(p), type, InverseFunctionalProperty) \mid (s, p, o) = t \in G\} \cup ((singletonPropertyOf, subPropertyOf, subPropertyOf))$.*

The Companion Property [4] is an attempt to reduce the number of unique properties that are generated by singleton properties. Its contextualization function can be defined as follows.

Definition 7 (Companion Property contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{cp} such that* $f_{cp}(c, G) = \{(s, \mathsf{ren}_1(p), o), (\mathsf{ren}_1(p), companionPropertyOf, p), (s, \mathsf{ren}_2(p), \mathsf{ren}_3(t)), (\mathsf{ren}_3(t), idPropertyOf, \mathsf{ren}_1(p)) \mid (s, p, o) = t \in G\} \cup (companionPropertyOf, subPropertyOf, subPropertyOf)$.

NdFluents [5] was created with the purpose of improving inference preservation. It consists in considering individuals as separate entities that exist according to each context. Its contextualization function can be defined as follows.

Definition 8 (NdFluents contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{ndf} such that* $f_{ndf}(c, G) = \{(\mathsf{ren}_c(s), p, \mathsf{ren}_c(o)), (\mathsf{ren}_c(s), type, \mathsf{ren}_c(contextualPart)), (\mathsf{ren}_c(o), type, \mathsf{ren}_c(contextualPart)), (\mathsf{ren}_c(ContextualPart), subClassOf, ContextualPart), (\mathsf{ren}_c(s), \mathsf{ren}_c(contextualPartOf), s), (\mathsf{ren}_c(o), \mathsf{ren}_c(contextualPartOf), o), (\mathsf{ren}_c(s), \mathsf{ren}_c(contextualExtent), (\mathsf{ren}_c(o), \mathsf{ren}_c(contextualExtent), \mathsf{ren}_c(t)), (\mathsf{ren}_c(t), type, \mathsf{ren}_c(Context)), \mid (s, p, o) = t \in G\} \cup \{(\mathsf{ren}_c(Context), subClassOf, Context), (\mathsf{ren}_c(contextualExtent), subPropertyOf, contextualExtent), (\mathsf{ren}_c(contextualPartOf), subPropertyOf, contextualPartOf)\}$.

NdTerms [10] is a generalization of NdFluents that has been specifically studied in the context of Description Logics. It extends the approach to every term in the ontology instead of just the individuals. If the initial ontology is in OWL, then it can be represented in RDF, and the contextualization could be expressed according to the above Def. 1. However, it also introduces several notions that are required to ensure other desirable properties, that are fairly long to give in detail here. So we do not formalize the related contextualization function.

In addition to these approaches, there are diverse ways to express context with extensions of the syntax of RDF. Most prominently Named graphs [3], RDF* [6], and Notation 3 [1] are three such cases. Named graphs extend the syntax of RDF by adding a fourth element to the triples. RDF* extends RDF with the possibility to use an RDF* triple in subject and object positions of another RDF* triple. Notation-3 extends this further with more constructs, such as rule definitions, variable quantifications. These approaches do not conform to the definition of contextualization function, since the result of their transformation is not an RDF graph. Hence, they will be kept out of scope for comparisons in the rest of the paper.

4 The NdProperties Ontology and Contextualization Function

NdFluents [5] and NdTerms [10] are based on the idea if creating contextual terms (individuals for the former, all terms for the latter) that mirror the terms of the graph or ontology. They can be seen as concrete instantiations of a general approach in which a subset of the terms is contextualized. NdProperties is a new instantiation where the relations are contextualized. It creates a different property for each existing property in the graph and each different context we want to use to annotate the data. The ontology can be seen down below, and the contextualization function is shown in Definition 9.

```
Prefix( rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#> )
Prefix( owl:=<http://www.w3.org/2002/07/owl#>)
Prefix( ndp:=<http://w3id.org/nd/properties#> )
Ontology( <http://w3id.org/nd/properties>
      Declaration( Class( ndp:Context ) )
      Declaration( Class( ndp:ContextualProperty ) )
      SubClassOf( ndp:ContextualProperty rdf:Property ) )
      Declaration( Class( ndp:ContextualObjectProperty ) )
      SubClassOf( ndp:contextualObjectProperty ndp:ContextualProperty ) )
      SubClassOf( ndp:contextualObjectProperty owl:ObjectProperty ) )
      Declaration( Class( nd:ContextualDatatypeProperty ) )
      SubClassOf( ndp:contextualDatatypeProperty ndp:ContextualProperty ) )
      SubClassOf( ndp:contextualDatatypeProperty owl:DatatypeProperty ) )
      Declaration( ObjectProperty( ndp:contextualPropertyOf ) )
      ObjectPropertyDomain( ndp:contextualPropertyOf rdf:Property ) )
      ObjectPropertyRange( ndp:contextualPropertyOf ndp:ContextualProperty )
      Declaration( ObjectProperty( ndp:contextualExtent ) )
      ObjectPropertyDomain( ndp:contextualExtent ndp:ContextualProperty )
      ObjectPropertyRange( ndp:contextualExtent ndp:Context )
)
```

While the ontology can be used "as is", in order to represent more than one context it is necessary to be extended for each context (in a similar fashion as NdFluents [5]). In 1 we show the representation of a statement with temporal annotations using this ontology. The non-dashed parts belong to the temporal extension, while the dashed parts correspond to the NdProperties ontology plus other possible extensions. Other dimensions, such as provenance, can be modeled similarly to the temporal dimension by replacing the appropriate classes and properties. We will use the function ren in order to generate these terms in Definition 9.

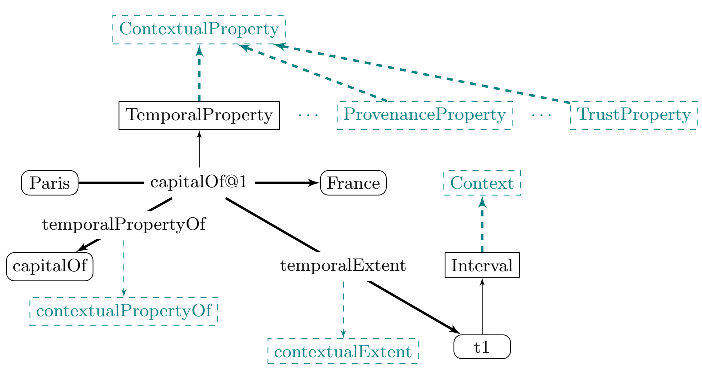

Fig. 1: Temporal Extension of NdProperties

Definition 9 (NdP$^-$ contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{NdP^-} such that* $f_{NdP^-}(c, G) =$

$$
\begin{aligned}
&\{(s, \mathsf{ren}_c(p), o) \mid (s, p, o) = t \in G\} \cup \\
&\{(\mathsf{ren}_c(p), type, \mathsf{ren}_c(ContextualProperty)) \mid p \in t \in G\} \cup \\
&\{(\mathsf{ren}_c(p), \mathsf{ren}_c(contexualPropertyOf), p) \mid p \in t \in G\} \cup \\
&\{\mathsf{ren}_c(p), \mathsf{ren}_c(contextualExtent), \mathsf{ren}_c(t)\} \\
&\{(\mathsf{ren}_c(ContextualProperty), subClassOf, ContextualProperty)\} \cup \\
&\{\mathsf{ren}_c(contextualPropertyOf), subPropertyOf, contextualPropertyOf\} \cup \\
&\{\mathsf{ren}_c contextualExtent, subPropertyOf, contextualExtent\} \cup \\
&\{\mathsf{ren}_c(t), type, \mathsf{ren}_c(Context)\} \cup \\
&\{\mathsf{ren}_c(Context), subClassOf, Context\}
\end{aligned}
$$

Note that the semantics of the property p are lost in f_{NdP^-} (*e.g.*, it's domain and range, or as a functional property, transitive property, etc.). However, it is also possible to have contextualization functions that retain totally or partially the semantics of p. For the sake of comparison, we will define a contextualization function f_{NdP^+}, in which we the contextual properties are described in an identical way as the original ones.

Definition 10 (NdP$^+$ contextualization function). *Let $c \in \mathcal{C}$ and $G \in 2^{\mathcal{T}}$ be a context and a graph respectively. We define the contextualization function f_{NdP^+} such that* $f_{NdP^+}(c, G) = f_{NdP^-}(c, G) \cup \{(\mathsf{ren}_c(p), q, r) \mid (p, q, r) = t \in G\}$

In the following section we will compare the inference preservation of the two NdProperties contextualization functions against the other approaches.

5 Reasoning with NdProperties

Following a similar approach as for NdFluents [5], we analyze for which rules from the pD* fragment of OWL [7] the contextualization functions preserve inferences. For each rule, we check if the contextualization function for each approach has *Inference Preservation* or *Non-Contextual Inference Preservation* wrt a graph that contains the corresponding triples. Following NdFluents evaluation, we apply the contextualization function only on triples that do not include

Table 1: Preserved D* entailments (P = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF	NdP$^-$	NdP$^+$
lg	$v\ p\ l$	$l \in L$	$v\ p\ b_l$	P	P	P	P	P	P
gl	$v\ p\ b_l$	$l \in L$	$v\ p\ l$	P	P	P	P	P	P
rdf1	$v\ p\ w$		p `type Property`			P	P	P	P
rdf2-D	$v\ p\ l$	$l = (s, a) \in L_D^+$	b_l `type` a	P	P	P	P	P	P
rdfs1	$v\ p\ l$	$l \in L_p$	b_l `type Literal`	P	P	P	P	P	P
rdfs2	p `domain` u $v\ p\ w$		v `type` u			P!			P!
rdfs3	p `range` u $v\ p\ w$	$w \in U \cup B$	w `type` u			P!			P!
rdfs4a	$v\ p\ w$		v `type Resource`	P	P	P	P	P	P
rdfs4b	$v\ p\ w$	$w \in U \cup B$	w `type Resource`	P	P	P	P	P	P
rdfs7x	p `subPropertyOf` q $v\ p\ w$	$q \in U \cup B$	$v\ q\ w$			P_{NC}!	P		

Table 2: Preserved P-Entailments (P = Rule Preservation, P_{NC} = Non-Contextual Rule Preservation, ! = Risk of undesirable inference)

Rule	Condition	Constraint	Conclusion	Reif.	N-Ary	S.P.	NdF	NdP$^-$	NdP$^+$
rdfp1	p `type FunctionalProperty` u p v u p w	$v \in U \cup B$	v `sameAs` w			P!			P!
rdfp2	p `type InverseFunctionalProperty` u p w v p w		v `sameAs` w			P!			P!
rdfp3	p `type SymmetricProperty` v p w	$w \in U \cup B$	w p v			P_{NC}!	P		P
rdfp4	p `type TransitiveProperty` u p v v p w		u p w			P_{NC}!	P		P
rdfp5a	v p w		v `sameAs` v	P	P	P	P	P	P
rdfp5b	v p w	$w \in U \cup B$	w `sameAs` w	P	P	P	P	P	P
rdfp8ax	p `inverseOf` q v p w	$w, q \in U \cup B$	w q v			P_{NC}!	P		P
rdfp8bx	p `inverseOf` q v q w	$w \in U \cup B$	w p v			P_{NC}!	P		P
rdfp11	u p v u `sameAs` u' v `sameAs` v'	$u' \in U \cup B$	u' p v'	P	P	P_{NC}!		P	P
rdfp14a	v `hasValue` w v `onProperty` p u p w		u `type` v			P!			
rdfp14bx	v `hasValue` w v `onProperty` p u `type` v	$p \in U \cup B$	u p w	P_{NC}	P_{NC}	P_{NC}	P_{NC}	P_{NC}	P_{NC}
rdfp15	v `someValuesFrom` w v `onProperty` p u p x x `type` w		u `type` v			P!			
rdfp16	v `allValuesFrom` w v `onProperty` p u `type` v u p x	$x \in U \cup B$	x `type` w			P!			

RDF, RDFS, or OWL vocabularies. The intuition behind this is that reification approaches are usually used to annotate data on relations between resources. Table 1 shows the D* (modified RDFS) entailment rules and rule preservations for each one of the approaches, whereas Table 2 presents the same information for P entailments (modified subset of OWL). Note that we remove those rows where both condition and conclusion include only triples with RDF, RDFS, or OWL vocabularies. A symbol P indicates that there is rule preservation for the corresponding approach, while a symbol P_{NC} denotes non-contextual rule preservation. A symbol ! means that it is possible to have undesirable inferences This happens when the contextualization function annotates at least one triple of the condition, and either we have non-contextual inference preservation, or we have inference preservation but the function does not modify the triple in the conclusion.

For other approaches the conclusions of NdFluents [5] hold: reification and n-ry relations show poor preservation of rules, where most of those rules could be considered tautologies. The Singleton Property provides a mixture of inference preservation and non-contextual inference preservation for all the rules, that can be useful when we want to annotate universally true facts, but it is not usable when we want to have contextual information that is not universally true. NdFluents, by contrast, has neither non-contextual rule preservation nor rule preservation that can lead to undesirable inferences for any rule. As for NdProperties, we can observe that NdP$^-$ behaves slightly better than reification and n-ary relations, having inference preservation for tow additional rules: `rdf1` and `rdfp11`. NdP$^+$ shows the potential of this approach: adding axioms to the contextual property leads to having inference preservation for the associated rules (which was not possible for NdFluents). However, in four out of eight cases it can lead to undesirable inferences, which means that what axioms should be added to the contextualization function needs to be carefully considered.

6 Conclusion and Future Work

NdProperties can be seen as a concrete instantiation of a general approach of contextualizing a subset of the terms in an ontology. Other instantiations that follow this approach are NdFluents [5] (where the individuals are contextualized) and NdTerms [10] (where all the terms are contextualized). In this paper, we have present an ontology to represent NdProperties in RDF and showed two possible contextualization function among many. We compare these contextualization functions against existing reification approaches in terms of inference preservation and see that it preserves more desirable entailments, with the additional advantage that can be tuned to preserve only a subset of desired inferences.

In the future, we plan to continue our research in the general approach of contextualizing terms, studying their instantiations and properties and how to combine and relate different contexts, and performing a systematic comparison of the different instantiations of the approach.

Acknowledgements: This work is supported by H2020 Marie Skłodowska-Curie ITN No 642795. Authors would like to thank Prof. Pierre Maret for his comments.

References

[1] Berners-Lee, T., Connolly, D.: Notation3 (N3): A readable RDF syntax. W#C member submission (2011).
[2] Brickley, D., Guha, R.: RDF Schema 1.1. W3C Recommendation (2014).
[3] Carroll, J.J., Bizer, C., Hayes, P.J., Stickler, P.: Named graphs. J. Web Sem. (2005).
[4] Frey, J., Müller, K., Hellmann, S., Rahm, E., Vidal, M.-E.: Evaluation of Metadata Representations in RDF stores. Sem. Web. J. (2017).
[5] Giménez-García, J.M., Zimmermann, A., Maret, P.: NdFluents: An Ontology for Annotated Statements with Inference Preservation. ESWC (2017).
[6] Hartig, O.: Foundations of RDF* and SPARQL* (An Alternative Approach to Statement-Level Metadata in RDF). AMW (2017).
[7] Horst, H.J.t.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. J.W.S. (2005).
[8] Nguyen, V., Bodenreider, O., Sheth, A.: Don't like RDF Reification?: Making Statements about Statements Using Singleton Property. WWW (2014).
[9] Noy, N., Rector, A., Hayes, P., Welty, C.: Defining N-Ary Relations on the Semantic Web. W3C Working Group (2006).
[10] Zimmermann, A., Giménez-García, J.M.: Contextualizing DL Axioms: Formalization, a New Approach, and Its Properties. WOMoCoE (2017).